

REMARKS

Claims 1-8 and 14-25 remain in this application. Claim 1 has been amended. Claims 9-13 that were withdrawn from consideration have been cancelled. Claims 14-25 have been added.

The Draftsperson objected to the drawings as not conforming to 37 C.F.R. § 1.84 or 1.152. In response, please find enclosed herewith a new set of drawings. The new set of drawings include the appropriate margins, lines, numbers, and letters.

Before responding to the Examiner's rejection based on the prior art, a brief description of the present application is provided. The present application is directed toward a method and apparatus for utilizing session resources on a shared (or thin) client network environment. Computers in a network environment can be categorized as two types: servers and clients. In addition, a client can be further understood to be a thin client (in contrast with a thick client or a full-featured workstation). A thin client or a Desktop Unit (DTU) is a small, stateless, "plug and work" desktop computer whose main function is to process all input and output for the user and to manage communication with at least one server. All other client processing for the user are concentrated on a group of client servers and shared amongst a community of DTUs. The group of client servers can be called a shared client (or a consolidated client) because, although the servers are often the equivalent of larger powerful server machines, they perform the traditional role of the traditional client in a traditional client/server architecture. In addition, the shared client is shared by a large number of DTUs (that are shared by an even larger number of users on the DTUs). The removal of the traditional client processing (e.g., state maintenance and computation power) from the DTU (or thin client) into the shared client servers permits simplification of the DTU in the network because software and hardware for performing these tasks are not needed at the DTU.

Because the DTUs are stateless (i.e., devices that process information without any knowledge of previous/subsequent information), a user's interaction with the

network is managed using a persistent user session and the interaction can be instantly sent to any DTU within the network. That is, a user can be in the middle of a user session (associated with one or more services from one or more servers) on one DTU, move to another DTU, and then resume the user session exactly where the user left off. Similarly, if a DTU fails, a user can move from the failed DTU to another DTU without losing any work.

In one embodiment of the present invention, a DTU initiates a connection with any one of a plurality of available servers (e.g., a first server) in a group server environment. The DTU presents a token (which is not an IP address, but a logical identity of a user of the DTU) to the connected server. When the token is presented to the server, the server communicates with the other servers that are part of the group server environment to find one or more user sessions (within the group server environment) that are associated with the presented token. The DTU is then redirected to another server (e.g., a session server or a second server) having the one or more sessions. That is, the servers in the group server environment can communicate among each other to establish knowledge within each server of the connectivity of the servers so that a DTU, for example, can be redirected to connect to the most recently accessed session. Thus, the present invention provides a plurality of servers in a grouped server environment that are self-organizing, with no master component, and, hence, no single point of failure.

To put it another way, the present invention is oriented around a persistent session so that the loss of any persistent user information can be avoided. That is, if a session hosting server fails, the DTU using that server switches to an alternate session server for hosting the session.

The Applicants have amended Claim 1 and added new Claims 14-25 to clarify certain features of the subject matter being claimed.

The Examiner rejected Claims 1-2 under 35 U.S.C. § 103(a) as being unpatentable over Peacock. In addition, Claims 3-8 are rejected under 35 U.S.C. §

103(a) over Peacock in view of DeBettencourt. These rejections are respectfully traversed.

Peacock is directed to a method for solving the problem of a client that is attempting to find a uniquely named server whose Internet IP address is dynamically allocated. The method disclosed in Peacock explicitly calls out for the use of an Internet Protocol (IP) broadcast (sent from a client) to find a server whose IP address has changed. By contrast, the present invention involves a method for providing a way for a plurality of servers within a group to communicate with each other. The communication is utilized to distribute information (i.e., to create knowledge) within each server of the connectivity of the surrounding networks and to establish a list of participating servers in the group. This communication is done in a self-organizing manner, with no master component, and, hence, no single point of failure. A client (e.g., a DTU) in the present invention only seeks to connect to any one of the available servers in the group. The servers within the group redirect the client from, for example, a first session hosting server to a second session hosting server. Accordingly, the above differences between Peacock and the present invention show that the present invention is patentably distinguishable over Peacock.

Moreover, the present invention is not obviousness over Peacock because the problem being solved by Peacock is completely different. In Peacock, the client is attempting to find a uniquely named server whose IP address is dynamically allocated. In the present invention, a client (e.g., a DTU) seeks only to connect to any one of the available servers. The client of the present invention then presents a token, which is not an IP address, but the logical identity of the user of the client computer. When the token is presented, the server communicates with the other servers that are part of the group to find one or more sessions that are associated with the presented token. This is very different from a client that is only looking for an IP address corresponding to a host name of a server computer in that: a) the token associated with a user session is not the name of a physical computer; b) the result of the query of the present invention

to another server is not an IP address, but the location and characteristics of the user session that might exist on that server; and c) there can be a multiple numbers of user sessions associated with the token.

In addition, in the present invention, if no session exists, then one is created and the client device of the present invention is redirected to the newly created session. If a plurality of user sessions exist, the client device of the present invention is redirected to connect to the most recently accessed session. By contrast, Peacock only discloses a method of providing an IP address of a uniquely named server to a client so that the client can find the uniquely named server. Accordingly, because Peacock addresses a completely different problem than the present invention, the features in the present claims and not disclosed in Peacock can not have been obvious over Peacock under 35 U.S.C. §103. A rejection under prior art cannot be based on the use of the features in the present claims in combination with Peacock. Such a rejection would be using hindsight in an attempt to reconstruct missing features in Peacock to reject the instant claims.

More specifically, with regard to amended Claim 1, the Applicants respectfully submit that Peacock fails to suggest or disclose a method of making a computational service available in a multiple server computing environment, the method comprising:

exchanging information between a plurality of servers;

initiating a connection between a client unit and a first server;

determining at said first server a location of a session on one of said plurality of servers; and

redirecting said client unit via said first server to a second server having said session.

(Emphasis in bold added.)

Claims 2-8 should be allowable for at least the reasons that they each depend (directly or indirectly) from base Claim 1.

In addition, with respect to the rejections of Claims 3-8 over Peacock in view of DeBettencourt, the claims should each be independently allowable because DeBettencourt only teaches the use of a "manager" or "interceptor" component, which represent a single point of failure. By contrast, an objective of the present invention is to not have a single point of failure. That is, the present invention does not have a "manager" or "interceptor" because the servers of the present invention are self-organizing, with no master component, and, hence, no single point of failure.

Moreover, the present invention does not require a configuration database for representing the configuration of the group as is required by the teaching in DeBettencourt. The group of session servers of the present invention are completely self-configuring in such a way that the loss of any single server does not affect the operation of the others. For example, once a hash key (which may not even be required if the security feature is not enabled) is set on a new server and the server is turned on, this server sends out its configuration information to all the other servers in the group and receives information from the other servers in the group. In this way, all servers construct a description of the server group that they use to query for existing sessions.

Furthermore, it should be noted that DeBettencourt discloses nothing more than a conventional web server. The service model of a thin client session server of the present invention and a web server are completely different. Web services and connections using a typical web server are short, transaction-oriented interactions, with no updating of state on the server. By contrast, thin client sessions persist over a long time, and the session server contains the entire state of the user's session. In the present invention, the DTU (or thin client) is a stateless device having no state stored in the client. For example, unlike the service mode proposed in DeBettencourt, a user on

a client of the present invention can power-off the client, and the user can reconnect to the session (on the same or another client) and pick up exactly where the user left off, down to the position of the cursor on the screen.

Accordingly, the above differences between the present invention and DeBettencourt show that there is no motivation or teaching to use DeBettencourt (whether alone or in combination with Peacock) to reject the recitations of the present dependent Claims 3-8. Regardless, DeBettencourt does not make up for the deficiencies in Peacock cited above.

More specifically, with respect to the rejection of dependent Claim 4, the Applicants respectfully submit that DeBettencourt only discloses that states can be generated and stored (DeBettencourt: col 5/lines 25-30) and that a web server will be selected based on web server availability and load, administrator's changes, and application or web server start and shut down actions (DeBettencourt: col 7/lines 7-18). It does not disclose or suggest (whether alone or in combination with Peacock) that redirection to a particular web server is based on a "token" (or is even based on a "cookie") associated with a web transaction, as the Examiner alleges. In addition, there is no teaching in DeBettencourt of how a state of a user, or even if the state of the user, is used to select a particular web server. Thus, Claim 4 should be allowable for this additional reason.

New Claims 14-25 have been added to clarify certain additional aspects of the subject matter being claimed. The limitations in these new claims are not disclosed in or suggested by the cited references currently used to reject Claims 1-8 (whether alone or in combination). Indeed, Peacock and DeBettencourt (whether alone or in combination) are totally unconcerned with implementing server redundancy and client redirection in a network system in a server failure situation that is oriented around a persistent session so that the lost of permanent user data can be avoided.

Serial No. 09/513,015

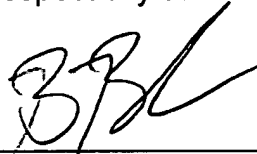
June 24, 2003

Page 12

In view of the foregoing, the Applicants respectfully submit that Claims 1-8 and 14-25 are in condition for allowance. Reconsideration and withdrawal of the rejections is respectfully requested, and a timely Notice of Allowability is solicited. To the extent it would be helpful to placing this application in condition for allowance, the Applicants encourage the Examiner to contact the undersigned counsel and conduct a telephonic interview.

While the Applicants believe that no fees are due in connection with the filing of this paper, the Commissioner is authorized to charge any shortage in the fees, including extension of time fees, to Deposit Account No. 50-0639.

Respectfully submitted,



Brian M. Berliner
Attorney for Applicants
Registration No. 34,549

Date: June 24, 2003

O'MELVENY & MYERS LLP
400 South Hope Street
Los Angeles, CA 90071-2899
Telephone: (213) 430-6000

Enclosure: Proposed Replacement Drawings (Figs. 1-9C)